My idea by Stefan Auburn

Use this to summarize your idea, plan it using sketches, notes and pseudocode as needed

|  |
| --- |
| Simply put create the arcade game asteroids but the ship can only fire 3 bullets and will only be given more bullets when the bullet either collides with an asteroid or collides with the ship (score will be determined with a timer as highest time is better score). To do this the game needs to have randomly placed asteroids that will respawn, a UI that shows how many bullets the ship can currently fire, a movement controls identical to the game asteroids using Pvectors and removing friction to create the feeling of controlling a ship in space and the ability to rotate and move forward in that direction. |

Where will the inventory skills be demonstrated? List every one to be sure you’ve included them.

|  |  |  |  |
| --- | --- | --- | --- |
| Demonstrate (one command from) each numbered item 1 – 43 (taken from skills inventory instructions)  *Shapes*  **1. ellipse** (will create asteroids and bullet), **rect and triangle** (will create the Ship),  **2. fill** (determines color of asteroids, ships and bullet), **stroke** (determines color of outline which will be used on asteroids, ships, and bullets)  **3. Modes: CORNER,** (center for sprites because it makes the math easier for me to understand)  *System*  **4. setup()** (will setup the game’s size as well as initial values for the game like player score and will be reset by the function that sends the player back to the title screen), **draw()** (will be used to make sure positions of sprites are updated when game is active)  **5. random()** (will be used to randomly place the asteroid on screen by having random x and y values) **background()** (will determine the background color/image as black)  **6. dist()** (will determine distance from asteroid to see if ship or bullets have collided) **constrain()** (will be used to ensure that only 3 bullets are allowed)  **7. keyPressed()** (will be used for the controls of wsad and arcade) , **keyReleased()** (will be used to stop movement when keys are released), **mousePressed()** (if gameplayactive is true instantiate a bullet),  **8. increment operators: ++** (going to be used in arrays of making asteroids), **+=** (will add to the score), **--** (subtract bullets from bullet array that allows only 3 bullets to be fired at once),  **9. declare and use a local variable** (example would be asteroidsize which is created and called in Asteroid object)  **10. declare and use a global variable** (example would be gameplayactive which is a variable that affects and possibly resets every objects value)  *Debugging*  **11. println** (will be used to test functions and see if they are running but will be commented out on final product as it will be finished and no longer to test functions example is seeing if the bullet array is running properly)  *Control flow*  **12. conditional statements: if, else if, else** (all will be used a lot, but examples include: gameplayactive if true starts the game if false (or “else”) stays on the title screen and will use many else if for determining which of WSAD is being pressed on the controls)  **13. Boolean expressions: ==, >=, <=, >, <, !=** (will be used a lot but examples include: is key that is pressed == to w, a, s, or d as well as is asteroid or ship or bullet >= the screen size or <= zero and if so will cause position to change to create loop like in asteroids)  **14. Logical operators: ||** (will be used to allow wsad controls *or* WSAD) **&&** (if key pressed and ontop of button)  **15. switch statement** (will determine and switch out the controls for wsad or arcade controls)  *Loops*  **16. for loop** (use for loops and arrays to create a specified number of asteroids), **while loop** (ship and asteroids only move while the game is active)  **17. A nested loop** (use nested loop to create stars in the background by making a loop that makes a pattern of stars in a horizontal line then put that loop inside another loop that repeats it to make a vertical and horizontal pattern)  **18. break()** (if using arcade gameplay it breaks wsad controls and vice versa)  **19. What’s the difference between a for loop and a while loop?** (For loop loops a specific number of times which is specified in the loop statement and while loop loops infinitely as long as the conditions are met and stops when they are broken)  *Functions*  **20. Declare & call a function with no parameters and no return type** (void makeship will call the ship/player but relies on variables that are changed in other functions)  **21. Declare & call a function with a return type** (the control function as both arcade controls and wsad controls change variables)  **22. What’s the difference between parameters and arguments?** Parameters are declared at the beginning of the function and argument is the value of the variable when the parameter is modified in the function.  **23. Pass by copy (value): declare and use a function that takes int, float, etc as an argument** (will be used to solidify coordinates of the ship)  **24. Pass by reference (objects): declare and use a function that takes an object as an argument** (will be used to update the score)  *Classes/objects*  **25. What’s the difference between a class and an object?** Objects are the entity that gets called and class are the blueprints that the code follows.  **26. What is a constructor function? What does it do and when?** Function that when actived instantiates the called object and fllows the blueprints of the class.  **27. Why should each class have its own tab in Processing?** Good for organization and illustrates that each object is it’s own thing that is separate from the main program unless called upon  **28. Write a class with a constructor function** (class asteroid sets asteroids variables and calls it into existence)  **29. Use the keyword new to instantiate an object** (new bullet and new asteroids for when asteroids gets replaced after getting destroyed)  **30. Write a constructor function with parameters** (constructing and spawning the asteroids within the gamespace)  *Lists*  **31. What’s the difference between an array and an ArrayList?** Arrays have a defined size (think the bullet list in trifighter being limited to only 3) and arraylists don’t (think like snake where you can continuously grow the snake)  32. **Why would you want to go through a list backwards, decrementing the index**? So you can work backwards in the array and check the most recent instantiations to the oldest instantiations  **33. Initialize and populate an array** (will be used spawning in the asteroids)  **34.** **Initialize and populate an ArrayList** (will be used for the bullets)  **35. Manage a set of objects with an array or ArrayList** (will be used for managing the asteroids and figuring out their position and velocity)  **36. Use an ArrayList method: size**() (stop creating asteroids when size() of list is greater than [insert number of asteroids that will make the game fun but not too difficult]),  Vectors  **37. When should you use PVector instead of float variables?** Simplifies the amount of floats and makes it easier to program velocity and position  **38.** **Use the PVector class** (yes will be used by the asteroids, ship and bullet)  **39. Do some basic physics: use position, velocity, and acceleration vectors** (all necessary for the ships and bullets as they will all have the ability to accelerate and have its position determined by the velocity)  **40. Find the direction and distance between two points** (will be used to determine distance between bullet and asteroid to see if they will collide and same with ship and asteroid)  **41. Create a random 2D vector** (will be used creating an asteroid whose position and velocity (speed and direction) is random)  42. **What is a normalized vector, why is it useful**? Sets vector to a length of 1 so direction position and speed can be easily determined  **43. Using the Processing documentation look up a method in the PVector class that’s new to you and use it in your code.** (copy() Pvector so I can have bullet copy the position of the player ship)  *Nice to Know* ***(optional)***  **44. ~~Use a timer~~** (no timer but stopwatch as it counts up and not down)  **45.** **Switch between “game states” (eg grounded/jumping) using conditional statements** (will be determining if “gameplayactive” is true or not)  **46. Make a button ~~or toggle switch~~ with a roll-over highlight (color or size change)** (buttons on main menu will do just that)  **47. ~~Create a drag & drop object~~** (no thank you)  **48. Do animation with images (spritesheet or individual files)** (may include a sprite change for explosion when ship crashes)  **49. Use collision detection between objects** (yes will check if asteroids, ships, bullet)  **50. Use the Game Control Plus controller library to get joystick or gamepad input** (I do want the game to run on both a computer and an arcade so I will follow the arcade instructions on slate to map the controls) | | | |
| **Milestone 1** | **Milestone 2** | **Milestone 3** | **Milestone 4** |
| What will I deliver?  draw ship, create menu, draw stars, and transition menu to gameplay (even if currently just houses a drawn ship), | Make ship movement, draw asteroids. Make controls for the ship movement, draw bullet, make asteroid movement, | You are strongly encouraged to deliver your finished game at Milestone 3.  create score,  Lock controls to arcade or wsad based on input, reset ship, collision, finish the bullet firing, make collision with bullets, ships, and asteroids, | Debug and fix anything that slips through the cracks. remove debug function after. |
| Which inventory skills will this demonstrate? List them.  purple means completed. |  |  |  |
| rectangles, triangles 1. | 41. Create a random 2D vector |  |  |
| Fill, stroke 2. | 24. Pass by reference (objects): | 15. switch statement |  |
| 30. Write a constructor function with parameters |  |  |  |
| Mode CENTER,. 3. | Initialize and populate Arraylist 34. | 18. break() |  |
| 4. setup(), draw() | Use an ArrayList method: size() 36. | 43. new pvector method |  |
| 10. declare and use a global variable | Manage a set of objects with an array or ArrayList 35. | 40. Find the direction and distance between two points |  |
| 13. Boolean expressions | 33. Initialize and populate an array | 23. Pass by copy (value): |  |
| 12. conditional statements | 28. Write a class with a constructor function |  |  |
| 14. Logical operators: | 8. increment operators: | 6. dist() |  |
| keyPressed 7. | 9. declare and use a local variable |  |  |
| 20. Declare & call a function with no parameters and no return type | 29. Use the keyword new to instantiate an object |  |  |
| 5. background() | 38. Use the PVector class |  |  |
| For loop 16. | 21. Declare & call a function with a return type |  |  |
| A nested loop 17. |  |  |  |
| 11. println | 39. Do some basic physics |  |  |
| **You should deliver approx. 10 skills at this milestone.** Done | **You should deliver approx. 10 skills at this milestone.** Will do | **You must deliver 30 inventory skills by this milestone.** understood |  |